Attorney Docket No.: 42390.P8104 Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:

Carl M. Ellison, et al.

Serial No.: 09/540,946

Filed: March 21, 2000

For: Protecting Software Environment In Isolated Execution

Art Unit: 2134

Examiner: Heneghan, Matthew

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313 on:

5-11-2005
Date of Deposit

Gayle Bekish
Name of Person Mailing Correspondence

Gayle B 5-11-2005
Signature Date

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

IN SUPPORT OF APPELLANT'S APPEAL

TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

Pursuant to Appellant's Notice of Appeal filed herewith, appellants hereby submit this Brief and fee under 37 C.F.R. § 1.17(c) in appeal of the Final Rejections as set forth in the Final Office Action ("FOA") dated February 23, 2005. Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences for allowance of the claims in the above-captioned patent application.

I. REAL PARTY IN INTEREST

The invention is assigned to Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95052.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellant's knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.

III. STATUS OF CLAIMS

Claims 1-48 are pending in the application. Claims 1-48 stand finally rejected. The rejections of independent Claim 1 and its dependent claims, independent Claim 13 and its dependent claims, independent Claim 25 and its dependent claims, as well as independent claim 37 and its independent claims are appealed.

IV. STATUS OF AMENDMENTS

The amendment filed September 2, 2004 has been entered, and is reflected in the listing of claims included in the Appendix (section VIII, below).

A request for minor claim amendments was recently filed under 37 C.F.R. § 1.116 on May 10, 2005. Appellant has not received any response to indicate whether those amendments have been entered. The amendments involved minor changes to put the claims in better form for consideration on appeal. The changes merely corrected claim dependencies and did not materially affect the substance of the claims, with regard to the grounds of rejection in the Final Office Action. This Appeal Brief therefore anticipates that the requested amendments will be entered.

A supplemental request for minor claim amendments was recently filed under 37 C.F.R. § 1.116 on May 11, 2005. Appellant has not received any response to indicate whether those amendments have been entered. The amendments involved minor changes to put the claims in better form for consideration on appeal. The changes merely corrected a typographical error in claim25 and deleted a redundant limitation in Claim 37. The amendments did not materially affect the substance of the claims, with regard to the

grounds of rejection in the Final Office Action. This Appeal Brief therefore anticipates that the requested amendments will be entered.

However, in case the amendments are not entered, the listing of claims provided in the Appendix (section VIII, below) includes the markings from the § 1.116 amendment requests, so that the prior version of the claims will be apparent. Such claims are marked with the parenthetical "(currently amended, first)" for those changes requested in the May 10 request and "(currently amended, second)" for those changes requested in the May 11 request.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Embodiments of the present invention relate to protection of a subset of a software environment. Each of the claims recites an OS nub 16.

Claim 1, for example, pertains to an apparatus comprising a key generator (e.g.: Fig. 2, 240) and a usage protector (e.g.: Fig. 2, 250). Regarding the key generator, claim 1 recites that it is to generate an operating system nub key (OSNK) unique to an operating system OS nub (e.g.: Fig. 2, 16). Claim 1 further recites that the OS nub is part of an operating system to run on a secure platform (e.g.; Fig. 2, 200). Claim 1 further recites "the OS nub being further associated with ring hierarchy level".

Regarding the usage protector, claim 1 further recites that it is coupled to the key generator to protect usage of a subset of a software environment (e.g.: Fig. 2, 210) using the OSNK.

Claim 13, for example, pertains to a method that comprises generating an operating system nub key (OSNK) unique to an operating system (OS) nub (e.g.: Fig. 12, 16) and protecting usage of a subset of the software environment using the OSNK.

Claim 13 further recites that the OS nub is part of an operating system to run on a secure platform (e.g.: Fig. 2, 200). Claim 13 further recites "the OS nub being further associated with ring hierarchy level".

Claim 25, for example, pertains to a computer program product having computer readable program code for generating an operating system nub key (OSNK) unique to an operating system (OS) nub (e.g.: Fig. 2, 16) and computer readable program code for protecting usage a subset of the software environment using the OSNK.

Regarding the OS nub, claim 25 further recites the OS nub being part of an operating system to run on a secure platform (e.g.: Fig. 2, 200) and also recites "the OS nub further being associated with a ring hierarchy level."

Claim 37, for example, pertains to a system comprising a processor (e.g.: Fig. 1C, 110), a storage (e.g.: Fig. 1C, 140) coupled to the processor, a key generator (e.g.: Fig. 2), and a usage protector (e.g.: Fig. 2, 250). Claim 37 further recites storing a subset (e.g.: Fig. 2, 230) of a software environment (e.g.: Fig. 2, 210).

Regarding the key generator, claim 37 further recites that the key generator is to generate an operating system nub key (OSNK) unique to an operating system (OS) nub (e.g.: Fig. 2, 16), the operating system nub being part of a software environment to run on a secure platform (e.g.: Fig. 2, 200). Claim 37 further recites the usage protector is coupled to the key generator to detect a subset of the software environment using the OSNK.

Claim 37 further recites "the operating system nub further being associated with a ring hierarchy level."

Additional variants are disclosed and represented in the dependent claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.

The issues for consideration on this appeal are:

A. Whether the Examiner erred in rejecting claims 1 – 48 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,421,006 to Jablon et al. (hereinafter "Jablon '006"); and

B. Whether the Examiner erred in rejecting claims 1 - 48 under 35 U.S.C. § 103(a) as being obvious over Jablon '006.

ARGUMENT

All Claims (1 – 48) are Patentable Over Jablon '006

A. Claims 1 – 48 are not Anticipated by Jablon '006.

Claims 1-48 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,421,006 to Jablon, et al. ("Jablon ''006). However, Jablon '006 fails to disclose or teach all limitations for each rejected claim. Accordingly, the Office Action fails to make a prima facie case of anticipation, and such rejections should be withdrawn.

(i) ***Regarding "the OS nub being further associated with a ring hierarchy level."***

Claims 1, 13, 25 and 37 each recite "the OS nub further being associated with a ring hierarchy level." The Final Office Action has failed to make out a prima facie case of anticipation regarding this limitation.

Jablon '006 discloses a method and device that prevents execution of corrupted programs at the time of system initialization (See Abstract; see also Col. 1, lines 10 – 12: "an improved method and device for assessing the integrity of computer software during the system initialization process").

The Jablon '006 method and device are particularly directed to systems that utilize the "ringless 'real addressing mode" of the DOS operating system. (Jablon '006, Col. 3, lines24-26; see also Jablon '006, col. 10, lines 19-21). That is, Jablon '006 is specifically directed away from protection rings and multi-ring architectures (see Col. 3,

lines 1 – 63) (Jablon '006 addresses systems running DOS, with ringless "real" addressing mode).

Jablon '006 acknowledges that multi-ring architectures were designed to "protect trusted operating systems from less-trusted applications." Col. 3, lines 2-3. UNIX, which uses a two-ring architecture, is used by Jablon '006 as another example of a system having "architectural strength." Col. 3, lines 11-15. Applicants respectfully, but ardently, disagree that Jablon states that UNIX "would benefit from the Jablon's [sic] invention." See FOA, p. 4, para. 7, citing Jablon, Col. 3, lines 12-23. Jablon '006 does admit that preventing unauthorized root access is a well-known security problem of UNIX systems. Col. 3, lines 20-23.

However, as the lengthy Background section of Jablon '006 makes clear, this type of multi-ring architecture system is not the type of system to which Jablon '006 is directed. Instead, Jablon '006 is directed toward ringless architectures for which '[t]here is no architectural constraint preventing any application form corrupting the rest of the system software." Col. 3, lines 24-28.

Jablon '006 reiterates is focus on ringless systems as the Background section progresses, stating a lines 50-53 of Col. 3 that: "[b]ecause PC DOS systems have no solid memory protection architecture, all writable storage areas of the machine are thus vulnerable to attack."

Finally, near the end of the Background section, Jablon '006 makes it clear that "**our invention**" is directed toward assessing the integrity of transient software that "allow[s] systems **without strong protection architecture** to nevertheless use strong security methods during system initialization." Col. 7, lines 21 – 24 (emphasis added).

The Final Office Action admits that "Jablon only discloses a ringless embodiment (using DOS)." FOA, p. 4, para. 7. Without citing any reference, the FOA goes on to make the unsupported assertion that "Jablon would clearly work with a ringed system." FOA, p. 4, para. 7. Such statement clearly falls short of the requirements for a prima facie case of anticipation.

"[F]or anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention ..." MPEP 706.02 (emphasis added). "The identical invention must be shown in as complete detail as contained in the ... claim." *Richardson v., Suzuki Motor Co.*, 868 F. 2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Jablon '006 simply fails to disclose every aspect of the rejected claims.

The Final Office Action states that "the invention may be used with other operating systems." FOA, p. 4, para. 7, citing Col. 10, lines 23-24. However, the cited portion of Jablon '006 does not disclose, suggest, nor teach "the OS nub further being associated with a ring hierarchy level." Instead, the cited portion of Jablon '006 at Col. 10, lines 23-24 merely states that other computer systems, besides an IBM-compatible PC system running a version of the DOS operating system, can be used for other embodiments. Presumably, such other computer systems used in other embodiments would also run one of the "ringless" operating systems to which Jablon '006 is directed (see discussion above). Such general statement does meet the burden of showing a prima facie case of anticipation regarding as complete detail as contained in the claim.

Granted, Jablon '006 includes another statement that the Final Office Action failed to cite, which also refers to alternative embodiments. The statement appears at Col. 1, lines 33 – 42:

As this invention provides the greatest benefit in personal computer systems, the IBM-compatible personal computer (herein referred to simply as the "PC") running the DOS operating system will be used as an example for much of this discussion. But the same benefits can be realized in *other computer operating systems* and hardware ..." (emphasis added)

Applicants note that this reference merely indicates, generally, that other operating systems may be used, but does not disclose in any way the use of a *multi-ring* architecture in any embodiment of the Jablon '006 invention. Again, this statement must be interpreted to mean that other ringless operating systems, to which the Jablon '006 invention is addressed, may be used. This statement, again, fails to disclose, suggest or teach "the OS nub further being associated with a ring hierarchy level."

Accordingly, Jablon '006 fails to disclose, teach, or suggest the limitations of the claims as presented. In particular, regarding Claims 1, 13, 25 and 37, Jablon '006 at least fails to disclose, teach or suggest the following limitation: "the OS nub further being associated with a ring hierarchy level." (Claim 1, in part). For at least this reason, independent claims 1, 13, 25 and 37 are allowable. Also for at least this reason, claims 2 – 12, 14 – 24, 26 – 36, and 38 – 48, which depend from the independent claims are also allowable.

(ii) ***Regarding "the OS nub being part of an operating system to run <u>on a secure platform</u>."***

Regarding Claims 1 - 48, the Final Office Action has also failed to make out a prima facie case of at least the following limitations: "the OS nub being part of an operating system to run on a secure platform" (Claims 1, 13, and 25, in part) and "the operating

system nub being part of a software environment to run on a secure platform" (Claim 37, in part). The Final Office Action states that all claims are anticipated by Jablon '006 but has failed to even attempt to make out a prima facie case as to such limitation.

The term "secure platform" appears nowhere in Jablon '006, and the FOA completely fails to make any argument that such limitation is disclosed in Jablon '006. Claims 1, 13, 25 and 37, as well as all claims that depend from Claims 1, 13, 25 and 37, are also allowable for at least this reason.

(iii) ***Regarding "the OS nub" and the "the operating system nub."***

The Final Office action also completely fails to make a prima facie case of anticipation regarding "the OS nub" (Claims 1, 13 and 25) and "the operating system nub" (Claim 37).

The Final Office Action states generally that "the system integrity scheme disclosed by Jablon includes the generation of a private key (using encryption) for each program level, including *parts of the operating system.*" FOA, p. 4, para. 7 (emphasis added). However, this general reference to "parts of the operating system" does not make a prima facie case of anticipation regarding "an OS nub". "The identical invention must be shown in as complete detail as contained in the ... claim." *Richardson v., Suzuki Motor Co.*, 868 F. 2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

The portions of Jablon '006 cited in the Final Office Action does not disclose "an OS nub." The Final Office Action cites Col. 16, lines 36 – 44 and Column 19, lines 28 – 44. Neither of the cited sections discloses, teaches, or suggests the limitation of "an OS nub."

At Col. 16, lines 36 – 44, Jablon '006 discloses that system software may be divided into trusted and untrusted categories. The boot record and the DOS may be part of the

trusted software. Such passage in no way discloses a nub of the operating system, but merely discloses that an entire operating system may be categorized as trusted software.

The second cited passage similarly fails to disclose the recited claim terms. That is, at Col. 19, lines 28 – 44, Jablon '006 discloses an alternative embodiment (see Fig. 7 of Jablon '006) to allow a subset of trusted software to be modified without propagating modification detection code changes. There is simply no disclosure, teaching, nor suggestion of an operating system nub.

Regarding Claim 1, Jablon '006 at least fails to disclose, teach or suggest the following limitation: "the OS nub further being associated with a ring hierarchy level". (Claim 1, in part). For at least this reason, Claim 1 is allowable. Also for at least this reason, Claims 2 - 12, which depend from Claim 13, are also allowable.

Regarding Claim 13, Jablon '006 at least fails to disclose, teach or suggest the following limitation: "the OS nub further being associated with a ring hierarchy level". (Claim 13, in part). For at least this reason, Claim 13 is allowable. Also for at least this reason, Claims 14 – 24, which depend from Claim 13, are also allowable.

Regarding Claim 25, Jablon '006 at least fails to disclose, teach or suggest the following limitation: "the OS nub further being associated with a ring hierarchy level." (Claim 25, in part). For at least this reason, Claim 25 is allowable. Also for at least this reason, Claims 26 – 36, which depend from Claim 25, are also allowable.

Regarding Claim 37, Jablon '006 at fails to disclose, teach or suggest the following limitation: "the operating system nub further being associated with a ring hierarchy level." (Claim 37, in part). For at least this reason, Claim 37 is allowable. Also for at least this reason, Claims 38 - 48, which depend from Claim 37, are also allowable.

B. Claims 1 – 48 are not rendered Obvious by Jablon '006.

Claims 1-48 stand rejected under 35 U.S.C. § 103ab) as being anticipated by U.S. Patent No. 5,421,006 to Jablon, et al. ("Jablon '006). However, the Office Action fails to make a prima facie case of obviousness, and such rejections should be withdrawn.

The legal requirements for a prima facie case of obviousness are clear. "The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness." MPEP § 2142. It is well established that *prima facie* obviousness is only established when three basic criteria are met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (MPEP 2144).

The Final Office Action has attempted to make out a prima facie case of obviousness for all claims 1-48 over a single reference, Jablon '006. The Final Office Action states that "it would have been obvious to one having ordinary skill in the art at the time the invention was made to implement the invention disclosed by Jablon in a ringed operating system, such as UNIX, as preventing root access is a well-known security problem of the system." FOA, page 5, para. 7.

Such rejection fails to make out a prima facie case of obviousness, at least because the prior art reference does not teach or suggest all the claims limitations for the rejected claims. At the very least, Jablon '006 fails to teach or suggest a "secure platform" (see section A(ii) above) and also fails to teach or suggest "the OS nub" and "the operating system nub" (see section A(iii) above). Applicants therefore strenuously

maintain that the Final Office Action has failed to present a prima facie case of obviousness because Jablon '006 does not teach or suggest all the claim limitations of Applicants' claims.

"When the references cited by the examiner fail to establish a prima facie case of obviousness, the rejection is improper and will be overturned." In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). Accordingly, all independent claims are in condition for allowance. For at least the foregoing reasons, all dependent claims are also in condition for allowance.

Appellant respectfully submits that all pending claims in this patent application are patentable, and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

If any fee insufficiency or overpayment is found, please charge any insufficiency or credit any overpayment to Deposit Account No. 02-2666.

Respectfully submitted,

Intel Corporation

Dated: May 11, 2005

/Shireen Irani Bacon/
Shireen Irani Bacon
Registration No. 40,494
512.314.0435

Attorney Phone Number: (512) 732-3917

Correspondence Address: Blakely Sokoloff Taylor & Zafman, LLP
12400 Wilshire Blvd

Seventh Floor
Los Angeles, California 90025-1026

VIII. CLAIMS APPENDIX

1. (previously presented) An apparatus comprising:

a key generator to generate an operating system nub key (OSNK) unique to an operating system (OS) nub, the OS nub being part of an operating system to run on a secure platform, the OS nub further being associated with a ring hierarchy level; and

a usage protector coupled to the key generator to protect usage of a subset of a software environment using the OSNK.

2. (original) The apparatus of claim 1 wherein the key generator comprises:

a combiner to combine an identification of the OS nub and a master binding key (BK0) of the secure platform, the combined identification and the BK0 corresponding to the OSNK.

3. (original) The apparatus of claim 2 wherein the identification is a hash value of one of the OS nub and a certificate representing the OS nub.

4. (original) The apparatus of claim 1 wherein the usage protector comprises:

an encryptor to encrypt the subset of the software environment using the OSNK.

5. (previously presented) The apparatus of claim 1 wherein the usage protector comprises:

a first encryptor to encrypt a first hash value of the subset of the software environment using the OSNK and to store the encrypted first hash value in a storage;

a second encryptor to encrypt a second hash value of the subset of the software environment using the OSNK; and

a comparator to compare result between the encrypted second hash value and the encrypted first hash value retrieved from the storage, the comparison result indicating if the subset of the software environment has been modified.

6. (previously presented) The apparatus of claim 1 wherein the usage protector comprises:

a first encryptor to encrypt a first hash value of the subset of the software environment using the OSNK and to store the encrypted first hash value in a storage;

a decryptor to decrypt the encrypted first hash value using the OSNK; and

a comparator to compare a second hash value and the decrypted first hash value retrieved from the storage, to determine if the subset of the software environment has been modified.

7. (original) The apparatus of claim 1 wherein the usage protector comprises:

a signature generator to generate a signature of the subset of the software environment using a private key; and

a signature verifier to verify the signature using a public key to protect usage of subset if the subset of the software environment has been modified.

8. (previously presented) The apparatus of claim 1 wherein the usage protector comprises:

a manifest generator to generate a manifest of the subset of the software environment, the manifest to describe the portion of the software environment;

a signature generator coupled to the manifest generator to generate a manifest signature of the manifest using a private key;

a decryptor to decrypt the private key using the OSNK;

a signature verifier coupled to the signature generator to verify the manifest signature using a public key; and

a manifest verifier coupled to the signature verifier to verify the manifest, the verified manifest to indicate if the subset of the software environment has been modified.

9. (previously presented) The apparatus of claim 1 wherein the secure platform is to use an isolated execution mode.

10. (previously presented) The apparatus of claim 1 wherein the software environment is one of a WINDOWS® operating system, a WINDOWS® 95 operating system, a WINDOWS® 98 operating system, a WINDOWS NT® operating system, and a WINDOWS® 2000 operating system.

11. (original) The apparatus of claim 9 wherein the subset of the software environment is a registry of an operating system.

12. (original) The apparatus of claim 2 wherein the BK0 is generated at random on a first invocation of a processor nub.

13. (previously presented) A method comprising:

generating an operating system nub key (OSNK) unique to an operating system (OS) nub, the OS nub being part of an operating system to run on a secure platform, the OS nub further being associated with a ring hierarchy level; and

protecting usage of a subset of the software environment using the OSNK.

14. (currently amended, first) The method of claim [[11]]13 wherein generating the OSNK comprises:

combining an identification of the OS nub and a master binding key (BK0) of the secure platform, the combined identification and the BK0 corresponding to the OSNK.

15. (currently amended, first) The method of claim [[12]]14 wherein the identification is a hash value of one of the OS nub and a certificate representing the OS nub.

16. (currently amended, first) The method of claim [[11]]13 wherein protecting usage comprises:

encrypting the subset of the software environment using the OSNK.

17. (currently amended, first) The method of claim [[11]]13 wherein protecting usage comprises:

encrypting a first hash value of the subset of the software environment by the OSNK and storing the encrypted first hash value in a storage;

encrypting a second hash value of the subset of the software environment using the OSNK; and

comparing the encrypted second hash value to the encrypted first hash value retrieved from the storage, the comparison result indicating if the subset of the software environment has been modified.

18. (currently amended, first) The method of claim [[11]]13 wherein protecting usage comprises:

encrypting a first hash value of the subset of the software environment by the OSNK and storing the encrypted first hash value in a storage;

decrypting the encrypted first hash value using the OSNK; and

comparing a second hash value to the decrypted first hash value retrieved from the storage, to determine if the subset of the software environment has been modified.

19. (currently amended, first) The method of claim [[11]]13 wherein protecting usage comprises:

generating a signature of the subset of the software environment using a private key; and

verifying the signature using a public key to detect if the subset of the software environment has been modified.

20. (currently amended, first) The method of claim [[11]]13 wherein detecting comprises:

generating a manifest of the subset of the software environment, the manifest describing the subset of the software environment;

generating a manifest signature of the manifest using a private key, the private key being decrypted using the OSNK;

verifying the encrypted manifest signature using a public key; and

verifying the manifest, the verified manifest indicating if the subset of the software environment has been modified.

21. (currently amended, first) The method of claim [[11]]13 wherein the secure platform uses an isolated execution mode.

22. (currently amended, first) The method of claim [[11]]13 wherein the software environment is one of a WINDOWS® operating system, a WINDOWS® 95 operating system, a WINDOWS® 98 operating system, a WINDOWS NT® operating system, and a WINDOWS® 2000 operating system.

23. (currently amended, first) The method of claim [[19]]21 wherein the subset of the software environment is a registry of the operating system.

24. (original) The method of claim 14, wherein the BK0 is generated at random on a first invocation of a processor nub.

25. (currently amended, second) A computer program product comprising:

a computer usable medium having computer program code embodied therein, the computer program product having:

computer readable program code for generating an operating system nub key (OSNK) unique to an operating system (OS) nub, the OS nub being part of an operating system to run on a secure platform, the OS nub further being associated with a ring hierarchy level; and

computer readable program code for protecting usage of a subset of the software environment using the OSNK.

26. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for generating the OSNK comprises:

computer readable program code for combining an identification of the OS nub and a master binding key (BK0) of the secure platform, the combined identification and the BK0 corresponding to the OSNK.

27. (currently amended, first) The computer program product of claim [[22]]26 wherein the identification is a hash value of one of the OS nub and a certificate representing the OS nub.

28. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for protecting usage comprises:

computer readable program code for encrypting the subset of the software environment using the OSNK.

29. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for protecting usage comprises:

computer readable program code for encrypting a first hash value of the subset of the software environment using the OSNK, the encrypted first hash value being stored in a storage;

computer readable program code for encrypting a second hash value of the subset of the software environment using the OSNK; and

computer readable program code for comparing the encrypted second hash value and the encrypted first hash value retrieved from the storage, the comparison result indicating if the subset of the software environment has been modified.

30. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for protecting usage comprises:

computer readable program code for encrypting a first hash value of the subset of the software environment using the OSNK, the encrypted first hash value being stored in a storage;

computer readable program code for decrypting the encrypted first hash value of the subset of the software environment using the OSNK; and

computer readable program code for comparing the second hash value and the decrypted first hash value retrieved from the storage, the comparison result indicating if the subset of the software environment has been modified.

31. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for protecting usage comprises:

computer readable program code for generating a signature of the subset of the software environment using a private key, the private key being decrypted using the OSNK; and

computer readable program code for verifying the signature using a public key to detect if the subset of the software environment has been modified.

32. (currently amended, first) The computer program product of claim [[21]]25 wherein the computer readable program code for protecting usage comprises:

computer readable program code for generating a manifest of the subset of the software environment, the manifest to describe the subset of the software environment;

computer readable program code for generating a manifest signature of the manifest using a private key that has been decrypted using the OSNK;

computer readable program code for verifying the manifest signature using a public key; and

computer readable program code for verifying the manifest, the verified manifest indicating if the subset of the software environment has been modified.

33. (currently amended, first) The computer program product of claim [[21]]25 wherein the secure platform is to use an isolated execution mode.

34. (currently amended, first) The computer program product of claim [[21]]25 wherein the software environment is one of a WINDOWS® operating system, a WINDOWS® 95 operating system, a WINDOWS® 98 operating system, a WINDOWS NT® operating system, and a WINDOWS® 2000 operating system.

35. (currently amended, first) The computer program product of claim [[29]]33 wherein the subset of the software environment is a registry of an operating system.

36. (original) The computer program product of claim 26 wherein the BK0 is generated at random on a first invocation of a processor nub.

37. (currently amended, second) A system comprising:

a processor;

a storage coupled to the processor, the storage storing a subset of a software environment; and

~~a usage protector comprising:~~

a key generator to generate an operating system nub key (OSNK) unique to an operating system (OS) nub, the operating system nub being part of a software environment to run on a secure platform, the operating system nub further being associated with a ring hierarchy level; [[and]]

[[a]] the usage protector coupled to the key generator to detect a subset of the software environment using the OSNK.

38. (currently amended, first) The system of claim [[31]]37 wherein the key generator comprises:

a combiner to combine an identification of the operating system nub and a master binding key (BK0) of the secure platform, the combined identification and BK0 corresponding to the OSNK.

39. (currently amended, first) The system of claim [[31]]38 wherein the identification is a hash value of one of the OS nub and a certificate representing the OS nub.

40. (currently amended, first) The system of claim [[31]]37 wherein the usage protector comprises:

an encryptor to encrypt the subset of the software environment using the OSNK.

41. (currently amended, first) The system of claim [[31]]37 wherein the usage protector comprises:

an encryptor to encrypt a first hash value of the subset of the software environment using the OSNK and to provide the encrypted first hash value to a storage; and

a comparator to compare the encrypted second hash value and the encrypted first hash value retrieved from the storage to determine if the subset of the software environment has been modified.

42. (currently amended, first) The system of claim [[31]]37 wherein the usage protector comprises:

an encryptor to encrypt a first hash value of the subset of the software environment using the OSNK and to store the encrypted first hash value in a storage; and

a comparator to compare the second hash value and the decrypted first hash value retrieved from the storage to determine if the subset of the software environment has been modified.

43. (currently amended, first) The system of claim [[31]]37 wherein the usage protector comprises:

a signature generator to generate a signature of the subset of the software environment using a private key and to decrypt the private key using the OSNK; and

a signature verifier to verify the encrypted signature using a public key to detect if the subset of the software environment has been modified.

44. (currently amended, first) The system of claim [[31]]37 wherein the usage protector comprises:

a manifest generator to generate a manifest of the subset of the OS, the manifest to describe the portion of the software environment;

a signature generator coupled to the manifest generator to generate a manifest signature of the manifest using a private key and to decrypt the private key using the OSNK;

a signature verifier coupled to the encryptor to verify the encrypted manifest signature using a public key; and a manifest verifier coupled to the signature verifier to verify the manifest to indicate if the subset of the software environment has been modified.

45. (currently amended, first) The system of claim [[31]]37 wherein the secure platform is to use an isolated execution mode.

46. (currently amended, first) The system of claim [[31]]37 wherein the software environment is one of a WINDOWS® operating system, a WINDOWS® 95 operating system, a WINDOWS® 98 operating system, a WINDOWS NT® operating system, and a WINDOWS® 2000 operating system.

47. (currently amended, first) The system of claim [[39]]45 wherein the subset of the software environment is a registry of an operating system.

48. (original) The system of claim 38 wherein the BK0 is generated at random on a first invocation of a processor nub.